

핵심 160. 관계 설정

잘못된 데이터의 입력을 사전에 방지, 여러 테이블 간 정보 연결하여 가져올 수 있도록 테이블 간 관계 정의
기본 테이블에서는 기본키 필드, 관련 테이블에서는 외래키 필드를 서로 대응시켜 관계 설정
기본키 필드와 외래키 필드는 데이터 형식이 같아야 하며 같은 종류의 정보가 들어 있어야 함
열려 있는 테이블에는 관계를 설정 X

핵심 161. 참조 무결성

관련된 테이블 간의 관계를 유지하고, 사용자가 실수로 변경 삭제하지 않도록 MS에서 사용하는 규칙.

외래키 키 값을 기본 테이블의 기본 필드값과 동일하게 유지하는 것이 제약조건.

'관계 편집' → '항상 참조 무결성 유지'

참조 무결성 설정 조건

- 기본 테이블에서 사용할 필드는 기본키이거나 고유 인덱스가 설정되어 있어야 함.

- 관계 설정에 사용될 두 테이블의 필드는 같아야 함.

- 모두 Access DB 테이블이어야 함.

참조 무결성의 강화 규칙

- 기본 테이블의 관계가 설정된 테이블에 일치하는 레코드가 존재할 때만 기본 테이블에서 기본키 바꿈 X

- 기본키 필드 값이 바뀌면 자동으로 관련 필드의 값이 모두 수정되도록 설정 O

핵심 162. 레코드 추가/삭제

레코드 추가

새로운 레코드를 추가하는 것으로, 항상 마지막에 추가. 지정된 필드 속성에 맞게 데이터 입력

레코드 삭제

여러 레코드를 삭제할 경우 레코드 선택기를 드래그하여 여러 개의 레코드를 선택한 후 삭제

삭제된 레코드는 복구 불가

핵심 163. 외부 데이터 가져오기 / 연결하기

가져오기

가져올 데이터를 사용해 새 테이블을 만들 수 있으며

데이터 구조가 일치할 경우 기존 테이블에 가져온 데이터 추가 가능.

원본 데이터는 변경 X, 가져온 데이터 변경해도 원본 데이터 영향 X

연결하기

원본 DB의 데이터(레코드)를 삭제하면 연결된 테이블의 레코드도 삭제.

연결된 테이블 삭제 가능, 삭제해도 원본 데이터 영향 X

핵심 164. 데이터 내보내기

DB 개체를 다른 응용프로그램에서 사용할 수 있도록 형식을 변경하여 출력하는 것.

제약조건, 관계, 인덱스 같은 속성은 내보낼 수 X

쿼리를 내보낼 경우 실행 결과가 저장, 결과보고서를 서식 있는 텍스트 파일로 내보낼 경우 데이터 표시

핵심 165. 단순 조회 질의 - 기본 구문

```
SELECT [DISTINCT] 필드 이름
FROM 테이블 이름
[WHERE 조건식]
```

여러 줄에 나누어 입력 가능. 마지막 ';' 입력해 SQL의 끝임을 알림.

'DISTINCT'는 중복되는 레코드는 검색 시 한 번만 표시

필드 이름: 테이블 모든 필드 검색 이름 대신 '*', 필드와 필드는 집표(.)로 구분

WHERE 조건식: 조건을 입력하여 특정 조건을 입력하여 특정 조건에 맞는 레코드만 검색할 때

2020.06.11 일

핵심 167. 그룹 지정

```
SELECT [DISTINCT] 필드 이름
FROM 테이블 이름
[WHERE 조건식]
[GROUP BY 필드 이름]
[HAVING 그룹조건식]
```

GROUP BY: 특정 필드를 기준으로 그룹화하여 검색할 때 사용.

HAVING은 그룹에 대한 조건을 지정할 때 사용.

WHERE는 개개의 레코드에 조건을 지정, HAVING은 그룹에 대한 조건 지정

핵심 168. 주요 함수

NOW: 현재 날짜와 시간을 표시

LEN(필드명): 필드의 문자열 길이를 반환.

ROUND(필드명, 소수자릿수): 필드의 숫자를 지정한 자릿수로 반올림.

INSTR(문자열, 찾는 문자): 문자열에서 찾는 문자 또는 문자열의 위치를 반환.

DATEPART: 지정된 날짜에서 형식에 제시된 값만 표시.

VAL(필드명): 문자열로 표시된 숫자를 숫자 값으로 반환.

LEFT(문자열, 자릿수): 왼쪽에서 주어진 자릿수만큼 추출

DATEVALUE(날짜): 텍스트 형식의 날짜를 일련번호로 변환

MONTHNAME(인수): 지정된 인수를 월로 나타낸 문자열로 반환.

TIMEVALUE(시간): 텍스트 형식의 시간을 일련번호로 변환.

X: AVG, SUM, COUNT, MIN, MAX는 그룹(합) 함수

AS문: 필드 이름이나 테이블 이름에 별명(Alias)을 지정할 때 사용하는 명령

핵심 169. 특수 연산자를 이용한 질의/하위 질의

IN: 필드의 값이 IN 연산자의 인수로 지정된 값과 같은 값을 사귀는 레코드만 검색

문법: WHERE 필드 또는 필드를 나타내는 식 IN(값1, 값2, ...)

Between: And 연산자를 수행한 결과와 같음

문법: WHERE 필드 또는 필드를 나타내는 식 Between 값1 AND 값2

LIKE: 대푯 문자를 이용해 필드의 값이 패턴과 일치하는 레코드만 검색

문법: WHERE 필드 또는 필드를 나타내는 식 LIKE '문자 패턴'

NOT: 필드의 값이 NOT 다음에 기술한 연산자의 결과에 포함되지 않는 레코드만 검색

문법: WHERE 필드 또는 필드를 나타내는 식 NOT 다음 연산

하위 질의: 선택 질의(SELECT), 실행 질의(INSERT, UPDATE, DELETE)

* 대푯문자: * 또는 % - 모든 문자를 대푯, ? 또는 - - 한 자리 문자를 대푯, # - 한 자의 숫자를 대푯

핵심 170. 조인

2개 이상의 테이블에 나뉜 저장된 정보를 1개의 테이블처럼 사용 위해 연결하는 방법을 정의

조인에 사용되는 기준 필드의 데이터 형식은 동일하거나 호환되어야 한다.

여러 개의 테이블을 조인할 경우 필드 이름 앞에 테이블 이름을 앞붙이(C)로 구분.

• 내부 조인 (Inner Join)

```
SELECT 필드이름
FROM 테이블이름1 INNER JOIN 테이블이름2
ON 테이블이름1.필드이름=테이블이름2.필드이름
```

가장 일반적인 조인의 형태, 조인된 필드가 일치하는 행만 결과에 포함.

• 왼쪽 외부 조인 (Left Join)

```
SELECT 필드이름
FROM 테이블이름1 LEFT JOIN 테이블이름2
ON 테이블이름1.필드이름=테이블이름2.필드이름
```

왼쪽 테이블에서는 모든 레코드를 포함하고, 오른쪽 테이블에는 조인된 필드가 일치하는 레코드만 포함.

• 오른쪽 외부 조인 (Right Join)

```
SELECT 필드이름
FROM 테이블이름1 RIGHT JOIN 테이블이름2
ON 테이블이름1.필드이름=테이블이름2.필드이름
```

오른쪽 테이블에서는 모든 레코드를 포함하고, 왼쪽 테이블에는 조인된 필드가 일치하는 레코드만 포함.

핵심 171. 실행 질의

삽입(INSERT)문

테이블에 레코드를 추가할 때 사용

• 직접 입력하여 추가하기

```
INSERT INTO 테이블이름(필드이름1, 필드이름2)
VALUES(필드값1, 필드값2, ...)
```

• 테이블로부터 추가하기

```
(INSERT INTO: 테이블이름(필드이름1, 필드이름2, ...))
SELECT 필드이름
FROM 테이블이름
WHERE 조건
```

여러 개 레코드 동시 추가 가능하지만, 한 번에 하나의 테이블만 추가 가능

레코드의 전체 필드를 추가할 경우 필드 이름 생략 가능

수정(UPDATE)문

```
UPDATE: 테이블이름
SET 필드이름=값1, 필드이름2=값2, ...
WHERE 조건
```

테이블의 필드 값을 변경할 때 사용

조건지정 X, 테이블 내의 전체 레코드에 대해 변경

조건지정 X, 한 번에 여러 레코드 필드 값 변경

삭제(DELETE)문

```
DELETE *
FROM 테이블이름
WHERE 조건
```

테이블 삭제할 때 사용, 삭제한 내용 되돌릴 수 X

조건 지정 X → 테이블 전체 레코드 삭제

핵심 172. 기타 질의

크로스 탭 질의: 테이블의 특정 필드의 연관성을 표시, 데이터 세트 행, 열로 나뉜

행 머리를 사용된 필드들 여러 개를 지정, 열 머리를 원하는 필드만

그룹화한 데이터에 대해서 계산 가능

대체법칙 질의: 쿼리를 실행하면 여러 방식을 입력받을 수 있는 대푯값이 나뉘는 질의

대체법칙을 적용할 필드의 조건 지정을 대괄호([])로 묶어 입력.

2가지 이상의 정보를 묶는 질의를 매치하여 검색 가능

동합(Union) 질의: 설치가 위하한 테이블 질의의 내용을 합쳐 하나의 테이블로 제작

같은 레코드는 한 번만 기록, 열(필드) 개수 다르다면 통합 X

```
SELECT 필드이름
FROM 테이블이름1
UNION SELECT 필드이름
FROM 테이블이름2
```